

# Windows 8

What it means for development decisions today and in the future

**by Daniel Jebaraj**

# Contents

Now that BUILD is history.....	3
What we now know about Windows 8.....	3
The basics.....	3
The user interface .....	3
Desktop applications.....	3
winRT—the new Windows runtime.....	4
C++ with component extensions—native code .....	4
.NET .....	5
HTML-JavaScript Metro applications .....	5
Porting existing applications.....	5
App store.....	6
How does Windows 8 affect development considerations today?.....	6
Windows Forms.....	6
Windows Presentation Foundation (WPF) .....	6
ASP.NET (Web Forms).....	6
ASP.NET MVC.....	7
Silverlight .....	7
Windows Phone 7.x platform.....	8
Syncfusion’s coming suite of winRT-based controls .....	8
Appendix A.....	9
Before we knew the details.....	9
About Syncfusion .....	10

# Now that BUILD is history

We took stock of our planning in light of the new information that is now available. We are happy to report that all of our planning around Windows 8 is intact.

We have prepared a summary of what we now know.

## What we now know about Windows 8

### *The basics*

- Windows 8 runs on different form factors ranging from iPad-sized tablets to very large screens.
- Windows 8 will run on ARM-based hardware in addition to x86-based hardware.
- ARM hardware is often seen on low-power, low-cost devices. The ability of Windows 8 to run on ARM will be critical to Windows 8 adoption on devices that are different from traditional PCs.
- Windows 8 is not going to run on phone-sized devices.

### *The user interface*

- Windows 8 offers the new Metro-style tile UI currently seen on Windows Phone 7 devices.
- Fast and fluid user experience is at the heart of the new Metro UI.
- User experience is immersive—when an application runs it takes over the complete screen. The rest of the operating system is not seen.
- There is special emphasis on quick switching between applications.
- Applications are suspended when they are not in scope.
- The new UI is touch friendly but certainly not limited to touch. A keyboard and mouse can be used to navigate the new UI, though touch feels more natural.

### *Desktop applications*

- Windows 8 is Windows. The desktop remains unchanged. There are likely to be changes in this area before Windows 8 ships, but we expect that key elements of the desktop will not change. Windows 8 essentially operates in “Metro UI” mode and “Desktop” mode.
- The desktop is not considered legacy. It is considered to be parallel to Metro. Certain applications will continue to make more sense under the desktop though they may have parallel Metro implementations.
- WPF continues to be the platform of choice for desktop applications.

- Win32 is not going away anytime soon. Win32-based applications, including Windows Forms applications, will continue to work under Windows 8.

### ***winRT—the new Windows runtime***

- winRT is the new platform for building Metro-style applications.
- winRT is supported only on Windows 8 at the moment.
  - It is possible that it will be available on other legacy Windows platforms (less likely) and the Windows Phone platform (highly likely) in the future.
- XAML can be used as before in the development of winRT applications (except when using HTML/JavaScript).
- winRT is not implemented using managed code.
- winRT is written in C++ and exposed as a set of classes using technology based on concepts or an implementation that closely resembles the component object model (COM).
- winRT objects are reference-counted (as were COM objects).
- It is possible to interact directly with winRT using C++ and raw COM-like code. If you are interested in details, please take a look at <http://www.interact-sw.co.uk/iangblog/2011/09/16/real-native-winrt>.
- Most users of winRT will not work with native winRT. Instead, Microsoft has designed what it refers to as “Projections.” Projections are friendly manifestations of winRT within your development environment of choice.
  - The C++ projection for instance makes it easy to use C++ semantics WRT object construction and use. It manages object instantiation—reference counting under the covers. Collection classes are also mapped.
  - The same is true for the .NET languages and JavaScript.
- Sections of winRT can be used from desktop applications.
- The winRT graphics stack is built on DirectX. DirectX is supported under Metro. Direct2d is also supported.
- GDI and GDI+ are not supported when writing Metro-style applications.

### ***C++ with component extensions—native code***

- C++ can be used to develop native winRT applications that have no dependency on .NET.
  - C++ code compiled against winRT produces 100% native code.
  - There is no managed layer. This is in contrast to managed C++.
  - Memory management for C++ component extensions is automatic. Reference counting happens under the covers.
- C++ component extensions can make use of ANSI C++. The API they expose however has to be comprised of component extension objects.

- The code that is generated can be seen as a higher level abstraction of COM clients—essentially virtual table lookup under the covers.
- You can get the performance benefits of native code while working with a high-level API that is object- oriented.

### ***.NET***

- .NET is definitely not going away. It is supported under winRT.
- If you build winRT applications using C# or VB.NET, you will be using a version of the .NET framework that works in tandem with winRT.
- Just as .NET acted as a layer between platform memory management and platform APIs with the desktop, it will continue to do so on top of winRT. The internal implementation details have changed but how you use .NET has not changed.
- The version of .NET that runs inside winRT applications does not offer the same API surface as the full desktop version. It is smaller and similar to the API that is available on Silverlight.
- There is a new version of the .NET framework, version 4.5, that contains incremental changes for Windows 8 development. It includes access to winRT functionality for Windows 8 development.

### ***HTML-JavaScript Metro applications***

- HTML-JavaScript is a fully supported environment to build applications for Windows 8.
- These applications can have complete access to winRT. They do not offer more than the XAML-based approach but are a viable way to share code between your web applications and Metro-style applications.

### ***Porting existing applications***

- Most Silverlight/WPF code can be ported over to winRT.
  - The porting of code from Silverlight/WPF to winRT may be easy but we have to keep in mind that the UI paradigm is quite different.
  - This will require a re-thinking of user experience when compared to the current desktop or Silverlight applications.
- Silverlight applications will continue to work on Windows 8 with one important caveat: Silverlight is not supported under the Metro-style implementation of Internet Explorer. It is not likely to ever be supported under this version of IE. It will continue to work under the desktop version of IE.
- The skills for developing Metro-style applications carry over from WPF and Silverlight.

### *App store*

- Microsoft will offer an Apple-style app store for Metro-style applications.
- Applications will need to be certified by Microsoft to be listed on the store.
- Licensing models are built into the store.
- Desktop applications can also be listed but will not be integrated into the store in the way Metro-style applications will be.

## How does Windows 8 affect development considerations today?

Response is based on your current primary development platform.

### Windows Forms

- Nothing changes WRT Windows Forms. Windows Forms applications will continue to be supported on Windows 8.
- Syncfusion will continue to enhance and maintain our current suite of Windows Forms controls and frameworks.
- If you plan to have a corresponding application deployed on Metro/winRT, consider integrating new functionality with WPF. A XAML-based UI can be more easily shared with a winRT-based UI.

### Windows Presentation Foundation (WPF)

- WPF will continue to be the platform of choice when it comes to rich desktop applications.
- Consider a Metro client if it makes sense for your business needs.
- XAML code and skills carry over to winRT development, but be prepared for a complete re-evaluation of user interaction with your application.

### ASP.NET (Web Forms)

- ASP.NET code will continue to run on .NET 4.x and will be enhanced and supported as earlier.
- Consider migration to ASP.NET MVC. Alternatively, consider adding new functionality using ASP.NET MVC.

- ASP.NET MVC will make it easier to share code with Metro applications written using HTML/JavaScript. ASP.NET MVC is also easier to work with when planning for mobile development.
- Syncfusion has published a whitepaper on why you should seriously consider ASP.NET MVC for all web platform work. It is available at [www.syncfusion.com/downloads/resources/whitepaper/aspnet-mvc](http://www.syncfusion.com/downloads/resources/whitepaper/aspnet-mvc).

## ASP.NET MVC

- If a Metro-style application makes sense for your needs, it may be possible to share code with Metro applications written using HTML/JavaScript.
- Additionally, with ASP.NET MVC it is quite easy to add support for a mobile version of your site.
- Syncfusion is also adding a complete suite of mobile controls supported on all major mobile browsers.
- Syncfusion is adding a number of ASP.NET MVC extensions and frameworks based on HTML 5.

## Silverlight

- Silverlight does not run inside the Metro-UI IE browser. You will still need to maintain a separate Metro- UI client.
- If you are starting a new project with Silverlight, you should reconsider your decision.
- Silverlight is a good choice only if deployment concerns are paramount (XBAP deployment is not as seamless as Silverlight). In all other scenarios, WPF will do better.
- You can stick to a subset of WPF and produce a UI that looks great and offers excellent performance while sharing code with its winRT twin.
- Post BUILD we believe that WPF is a much better choice for a desktop UI.
- If deployment is your primary concern please consider an ASP.NET MVC solution. There is excellent tooling support on Windows and you can be very productive. You get excellent deployment characteristics and a cross-platform mobile story.
- **Going forward, Silverlight only makes sense on legacy projects. It makes sense on very few new projects.**
- Syncfusion will keep adding functionality as well as supporting Silverlight.
  - Our XAML team is integrated; most functionality will continue to be added to both Silverlight and WPF.
  - Syncfusion has always maintained separate WPF and Silverlight codebases when it comes to solutions that required deep integration with WPF (such as the grid and

chart controls). These controls will be independently enhanced for newer versions of Silverlight and WPF.

## Windows Phone 7.x platform

- We firmly believe that market share for Windows Phone 7 is slated for a major increase during the next two years.
- Consider offering a Windows 8 Metro UI for your existing Windows Phone 7 applications.
  - Quite a bit of the code can be shared, and you will be able to reach a wider audience once Windows 8 becomes available.
- Syncfusion will continue to produce controls and frameworks on this platform.

## Syncfusion's coming suite of winRT-based controls

Syncfusion is building a new set of winRT controls for Windows 8 Metro-style applications. We will publish details of these controls in the coming weeks. If you are interested in these controls, please contact us and we will be happy to share a preview version as soon as it becomes available.



# Appendix A

## Before we knew the details

We had posted information on Windows 8 as we saw it, to our blog on September 8, 2011. Key parts of the original post are reproduced here for reference.

Excerpts from “Are We Really in the Dark about Windows 8?” posted to our blog on September 8, 2011

We know the following for sure:

- Windows 8 will run on different form factors ranging from iPad-sized tablets to very large screens.
- Windows 8 will have a different start menu system based on the tile UI offered on Windows Phone 7 devices.
- Windows 8 is unlikely to run on phone-sized devices. The Windows Phone 7.x OS exists for that purpose.
- The new UI will be touch friendly but certainly not limited to touch. You will not be discarding your mouse and keyboard anytime soon.
- There is almost certainly a new Windowing platform (the Jupiter framework) that will be built on a tightly OS-integrated XAML stack. The skills for developing on this platform will likely carry over from the other XAML-based platforms.
- Jupiter may also take a step toward the unification of rich client and web-based deployment models— WPF and Silverlight—not the technologies as such, but the model itself may be unified under one common code base. If your WPF or Silverlight application is structured well, there should be no trouble migrating to another XAML-based framework, should you decide to do so.
- .NET is definitely not going away. Just why would Microsoft suddenly dump one of the most successful application development platforms of all time?
- Silverlight applications will work forever on Windows even if future development dramatically slows down (there is little indication that this is the case). WPF applications will work for a long time on Windows. Considering that applications written in C++ or Visual Basic more than a decade ago still work on Windows 7 with only minor tweaks, I am not very concerned about this aspect.
- Everything we know about Silverlight and WPF will likely carry over to the new framework. We expect only minimal developer training to take advantage of the new framework.

- HTML 5 will be a viable way to build applications for Windows 8. This is true of Windows 7 and many other platforms today. Any updates to IE and Windows that make it easier to target and deploy HTML 5 applications will be a big win for everyone. Note: if you are a rich-client developer with no current interest in HTML 5, JavaScript, and jQuery, now is the time to head over to Amazon and catch up.

Windows 8 is an exciting release for Microsoft and everyone who works within their ecosystem. I am eager to see what Windows 8 has to offer, but given Microsoft's excellent track record with developers, not the least bit worried.

As a company we will continue to make solid investments on our web, desktop, and mobile products. If you are a customer, rest assured that whatever comes down the pipe in September, Syncfusion will be ready with controls and frameworks that will enable you to ship rock-solid applications on time.

## About Syncfusion

Syncfusion is the enterprise technology partner for developers on every Microsoft platform, delivering the broadest range of .NET components and controls coupled with a service-oriented approach throughout the entire application lifecycle. Offering the fastest high-performance Silverlight and WPF grid on the market and quarterly product updates – Syncfusion is trusted by developers worldwide for use in mission critical applications. The company's flexible licensing and 24/7 support meets the changing needs of enterprises across the globe. Founded in 2001 and headquartered in Research Triangle Park, North Carolina, Syncfusion has more than 7,000 customers including large financial institutions, Fortune 100 companies and global IT consultancies.

For questions about products, pricing, and licensing, please contact [sales@syncfusion.com](mailto:sales@syncfusion.com) or call 1.888.9DOTNET (US only) or +1 919.481.1974.